Animating Highly Constrained Deformable Head/Face Models Using Motion Capture

November 6, 2006

Marcus Schoo
msc61@student.canterbury.ac.nz
Department of Computer Science and Software Engineering
University of Canterbury, Christchurch, New Zealand

Supervisors:
Dr R Green
richard.green@canterbury.ac.nz
Dr R Mukundan
mukund@cosc.canterbury.ac.nz

Abstract

We investigate, implement and evaluate algorithms and techniques that attempt to solve three key challenges when animating highly constrained deformable models by way of optical motion capture. We present two techniques to solve the pose estimation problem for rigid bodies, one based upon 3 fiducial markers falling on a plane and another built upon the ARToolkit. Further, we compare three techniques for fiducial marker segmentation, namely HSV colour space segmentation, camera hardware filtering and UV illumination segmentation. Lastly, we propose a hybrid pupil tracking algorithm combining Haar face detection, anthropometric localisation, pattern matching and row vs column intensity histograms.

We test the performance of our pupil detection algorithm on a 285 frame video displaying a variety of gaze directions. Our hybrid approach performs well, resulting in a very low pixel error and a reasonable frame rate. The three segmentation approaches are tested on four 1000 frame and four 100 frame videos containing inherently different movement. All approaches show excellent segmentation accuracy but illustrate the importance of understanding the limitations of each technique prior to implementation. Both pose estimation techniques are implemented and tested on objects placed in a variety of poses. Both algorithms show good approximations of pose.

Keywords

Motion capture, markers, markerless, ARToolkit, ultra violet, robotics, constrained models, face tracking, head tracking, pupil tracking.

Acknowledgments

Formostly, I would like to sincerely thank Dr Richard Green and Dr Ramakrishnan Mukundan for their support, encouragement and enthusiasm. I would like to thank Amol Malla and Kon Zakharov for their discussions on the hybrid pupil tracking algorithm. Further, I would like to thank Jennifer Schoo for her comments, patience, inspiration and understanding. Thank you to Dr Mark Billinghurst for suggesting this project and introducing me to WowWee Robotics. Lastly, a big thank you to Guy Nickless, Max Bogue and all the team at WowWee Robotics for the opportunity to work on this project. Thank you for your understanding and support.

Contents

List of Figures

1 Introduction

Since the beginnings of cartoon animation artists have been attempting to capture movement in some usable form. In 1917 Max Fleischer patented the rotoscope, a device allowing animators to create motion by tracing over live action film footage [1]. Such tools allowed animators to create natural movements by mimicking the real world. Since that time, science has investigated the topic of motion capture extensively.

Gleicher [7] describes motion capture as a technique that,

> "creates a representation that distills the motion from the appearance; that it encodes the motion in a form that is suitable for the kinds of processing or analysis that we need to perform."

That is, we wish to develop algorithms and techniques that allow the essence of a motion to be recorded separately from the appearance of the entity that originally performed the motion. Many such techniques exist.

Originally, mechanical systems were developed that, when strapped to actors, recorded the movements or replicated the motion directly on another machine. Between 1980-1983 research at Simon Fraser University [26] in biomechanics developed mechanical devices to measure joint angle of a subject. Advantages of mechanical systems not shared by optical or magnetic systems, such as ease of outdoor use, are creating a revival of mechanical systems such as those now available from METAMotion [1]. Magnetic location systems [6] such as those available from Ascension [2] generate a magnetic field and have electromagnetic receivers placed on the object to be tracked. These receivers detect their position within the field and report back to a collation station via either wired or wireless means.

Perhaps the most prevalent motion capture technique is that of optical motion capture. Optical motion capture involves placing many (sometimes up to 40 or more) small fiducial markers, coated in a retroreflective material, to the object to be tracked. A collection of cameras (as many as 20) are now used to triangulate the position of each marker in 3-space. Such systems are effective however suffer from many drawbacks. High resolution and high frame rate requirements result in these systems requiring cameras that may cost up to US$1000 each. Total systems, such as those available from Vicon [3] can cost as much as US$200,000.

Such motion capture systems are prohibitively expensive to all but the most well funded of users. However, many applications of motion capture do not require such general technologies. Algorithms and techniques should be sought that take advantage of constraints provided by applications that are a subset of all motion capture applications. We consider several areas of motion capture as it pertains to the capture of movements of the human head and face.

Three point pose estimation is a technique whereby a rigid body's pose is estimated with the help of a single camera and 3 markers. Many optical motion capture systems identify the 3d position of a object by attaching 3 markers to it. Our technique differs from these as only a single 320 x 280 pixel colour camera is used, in contrast to commercial optical systems using up to twenty 4-megapixel cameras. The 3 point pose estimation technique attempts to find three angles describing the objects rotation around the x, y and z axes. To do this, trigonometric operations are performed upon the projections of the markers on the image plane.

[1] http://www.metamotion.com/gypsy-motion-capture-system/gypsy-motion-capture-system.htm
[2] http://www.ascension-tech.com/products/motionstarwireless.php
[3] http://www.vicon.com/

In comparison to the three point pose estimation technique we attempt to use the ARToolkit[14, 3, 2], an API normally used in the production of augmented reality applications, to gain rotational information via a single 320 x 280 pixel colour camera and an ARToolkit marker (see figure 4.3 on page 17). Such libraries provide cost effective access to pose estimation implementations that can be modified to satisfy motion capture requirements.

Many fiducial marker techniques, including those that capture head position and face feature positions, require techniques that can identify the position of each marker in the image plane. This problem is known as marker segmentation. Traditionally, marker centroids are identified by searching for local maximas in an image based on a HSV (Hue, Saturation, Value) colour space [32]. Often the robustness of marker segmentation is improved by forcing the actor to where non-reflective clothing and making markers out of retroreflective material. IR light sensitive cameras are also often used so that natural light does not interfer with the system[27].

We consider two further techniques to improve robustness of marker segmentation. Camera hardware filtering focuses on modifying the cameras properties so that only highly reflective markers register on the cameras image plane. We also consider the effectiveness of fluorescent markers under a lamp producing UVa light.

Distinct from optical motion capture and dominating recent research is computer vision based motion capture [29]. Rather than attaching markers to actors, computer vision based motion capture techniques attempt to detect edges, points, curves, shapes, textures etc to identify an object and its pose within a scene. With regard to head and face tracking the pupils play a particularly important role in applications. As well as animation, pupil position can be used for gaze based interfaces[13], driver safety aids[10] and student state sensing within tutoring systems.

Many possible solutions have been put forward to the problem of pupil tracking including approaches involving Haar cascades, thresholding, Hough transforms, templates and pattern matching. We present an algorithm that recursively narrows the region of interest from the full frame to the pupils. Our algorithm initially uses Haar-like features to identify the face before narrowing the region of interest further to the eyes based upon anthropometric ratios. The eyes themselves are identified through pattern matching before intensity histograms (a.k.a integral projection) are used to locate the pupil itself.

1.1 Outline

Chapter 2 of this report introduces work relevent to this report completed by other researchers. The main areas of investigation for this project are covered next in chapter 3, namely marker based head tracking in section 3.1, marker based face feature tracking in section 3.2 and markerless face feature tracking in section 3.3. Analysis of the investigated algorithms are described in chapter 4 and discussed in chapter 5. Final conclusions and suggestions for future work are given in chapter 6.

2 Background

In this chapter we outline some of the key prior research pertaining to the areas this report focuses on.

2.1 Head Pose

2.1.1 Head Pose Through 3 Markers

Segmentation is the process of finding the 2d coordinates of a marker in the image plane. One of the most common techniques for achieving this is by segmenting a range in the HSV colour space. Grant et al [9] describe a segmentation technique based on static colour (HSV) segmentation and compare this to a dynamic HSV segmentation technique. For the static colour technique, ranges are defined within which HSV values for each coloured marker in the captured image should fall. Their technique then creates a binary mask image for each marker colour by setting a mask image pixel to 1 if the respective captured image pixel falls within the HSV range for that marker colour and 0 otherwise. Morphological operators are used to filter the resulting masks so that only large blocks of the target colour are marked in the mask. A connected component algorithm is used to identify areas of colour forming connected components and a bounding rectangle is defined for each component. The largest rectangle is returned as the position of the found marker.

While this approach is simplistic, it may also be the preferred. Grant et al. discuss a second approach called Dynamic Colour Tracking. Such an approach varies the HSV colour ranges of the marker based upon a histogram representation of images colour state. That is, as the colour properties of the scene change due to changes in lighting etc, the ranges used to identify markers change also. This should allow ranges to be set more narrowly, reducing false positives without causing more false negatives to the static system. Unfortunately, Grant et al. report that the dynamic system has catastrophic effects on marker segmentation robustness, sometimes reducing accuracy down to as little as 18.6%.

Correspondence and pose estimation have received much attention over the last several decades. Moeslund and Granum [29] define pose estimation as "the process of identifying how a human body and/or individual limbs are configured in a given scene" and give a detailed overview of the most important papers.

Pose estimation, in motion capture, predominately takes advantage of multiple cameras giving 3d coordinates for each marker. Kirk et al. [15] describe a system that rebuilds skeletal information of a body from the 3d position of markers. Their technique uses a combination of spectral clustering and nonlinear optimisation to "determine the overall topology of the motion capture subject, the length of each segment in the skeleton, the assignment of markers to segments in the skeleton, and the relative location of each marker with respect to the segment to which it is assigned".

Single camera techniques also exist. Nanda and Fujimura [19] describe a technique specifically for head pose estimation. Their technique has two main characteristics. A learning algorithm is used in an attempt to make the system robust to cluttered scenes where segmentation is usually difficult. The learner used is a single hidden layer neural network using a sigmoid output function and a linear output layer. Secondly, while only a single camera is used, the camera is fitted with a infra-red range finder per pixel. As such, the camera returns not only colour information for each pixel but also, rather noisy, depth information. Nanda and Fujimura report significantly better

performance of their neural network based approach versus a benchmark Principal Component Analysis approach ($\approx 70\%$ vs $\approx 97\%$).

Our technique attempts to also gain a pose estimation with a single camera. However, no additional information, such as that supplied by a depth finder, is used in our system.

2.1.2 Head Pose Through ARToolKit

Optical motion capture systems have been used extensively in many fields to accurately capture human motion. That said, these systems are often prohibitively expensive, requiring expensive software and many cameras. In contrast, libraries such as the ARToolkit exist that, while not specifically designed for motion capture, contain optical pose estimation algorithms that may be able to be applied to motion capture.

Sementille et al [25] explore this idea by attempting to construct a full body motion capture system using a single camera and eleven ARToolkit markers mapping to 15 joints. Their technique focuses on tracking these markers through a particle filter [18] to drive a human avatar. While they claim "quite satisfactory" results, they give no empirical evidence for their technique so it becomes difficult to see directly how well ARToolkit is suitable for mocap.

To find a more rigorous empirical study of the accuracy of ARToolkit we turn to work completed by Malbezin et al. [17]. Their experiments focus on determining the accuracy of ARToolkit in determining the position, on an xy-plane, of an ARToolkit marker with respect to the camera. Clear results are given with the camera placed at many angles with respect to the marker, at various distance between one and three metres. An mean error, over 1000 samples, of 14mm was reported.

In this work we look to extend current work by considering accuracy in 3-space rather than 2-space.

2.2 Marker Face Feature Tracking

Initial efforts in human motion capture were focused on tracking rigid bodies such as arm, leg and torso movement. Such techniques however were poorly suited to tracking non-rigid, deformable objects such a the human face. As early as 1981, Platt and Badler [21] began investigating how to recognise and simulate the non-rigid movement in the human face. Their research focused on techniques that supported their larger project of recognising and simulating American Sign Language. They present a system of representing and simulating facial movement. They do not describe the capture process itself but focus instead on the type of features such a capture system would need to output so that an animation system could work effectively.

The dominant systems used for face motion capture today, such as systems available from Vicon etc, make use of fiducial markers. These systems have there history in research such as that completed by Williams [32]. Williams describes a technique for capturing facial expressions, in the form described by Platt and Badlet, by using small spherical markers coated in a retroreflective material. Retrorelective materials have the property that light falling upon them is always reflected directly back to the source of the light, irrespective of the angle of incidence. Figure 2.1 shows a camera's perspective of the markers as well as marks showing detected marker centroids. Williams avoids the point correspondence problem by having an operator initialise his system by identifying which centroid in the first frame corresponds to which area of the face. For subsequent frames a marker is found by finding the center of intensity of a window surrounding the centroids previous position using a running average.

Recently the Motion Picture Expert Group (MPEG) released the MPEG-4 standard. Rather than being just a video compression standard the MPEG-4 standard included support for 3D graphic model specifications. Recently, much motion capture research has focused on capturing face movement in a format that matches the MPEG-4 face feature point set defined as the Facial Definition Parameters (FDPs) [8]. Such a standard allows all face feature techniques to have a standard set of points which should be tracked. The full set of FDPs in given in figure 2.2.

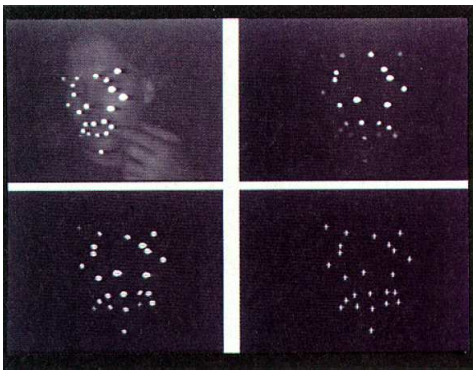


Figure 2.1: Images from Williams showing retroreflective markers and detected centroids[32].

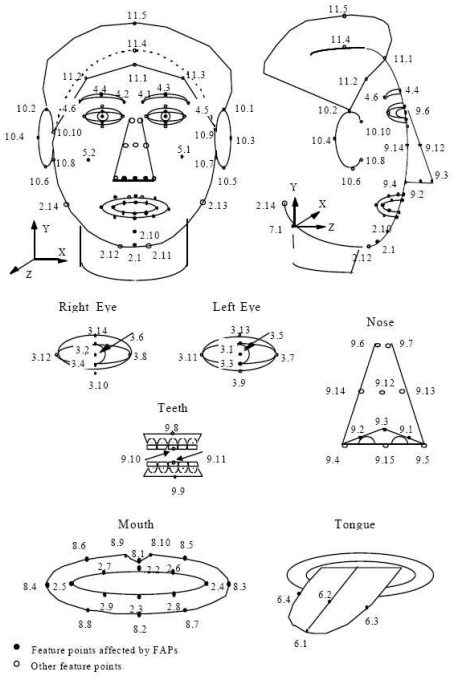


Figure 2.2: Facial Definition Parameters making up the MPEG4 standard[8].

2.3 Markerless Face Feature Tracking

Applications requiring the accurate identification of eye pupil position in a two dimensional image can be found in many areas, ranging from gaze based computer interfaces [4] to motion capture. Many possible solutions have been put forward to this problem including approaches involving Haar cascades, thresholding, Hough transforms, templates and pattern matching.

Kapoor and Picard [12] present pupil tracking by using the red-eye effect. Their pupil tracking technique makes up part of a larger gaze based input system. A IR sensitive camera is mounted under a computer monitor the subject is using. Further, IR LEDs shine light upon the user which is reflected and perceived by the camera. As with regular light, the cornea of the eye reflects IR light at a much higher rate than the surrounding area. As such, when the systems LEDs are active, the pupils of the subject glow white. Kapoor and Picard use one image taken with the light on and subtract from it an image take with the LEDs off. This has the effect of subtracting from the frame everything other than the two bright pupils (see 2.3. Finding the pupils in such

an uncluttered scene becomes a simple matter of intensity segmentation. The main problem with Kapoor and Picard's technique is that specialist equipment is needed. We prefer a system that makes use only of commonly available comsumer electronics.

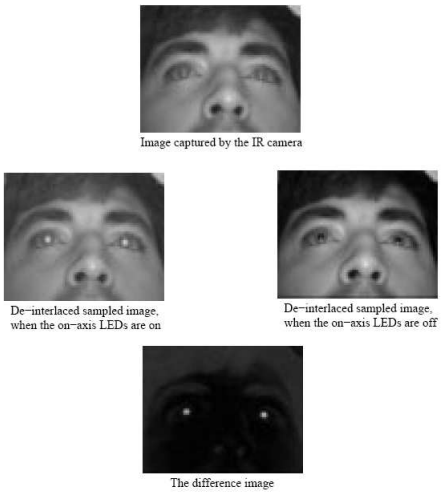


Figure 2.3: Kapoor and Picard's method for pupil detection[12].

Tian et al [28] propose a dual state, convergent tracking approach to determine many eye parameters. A deformable template or model is used to identify features in the face. Templates provide a relationship between facial features and result in a region of interest for a particular feature being narrowed. The template may then be deformed slightly to fix the position of the feature. For feature tracking over multiple frames Tian et al. use a modification of the Lucas-Kanade tracking algorithm. What is most interesting about their approach is that they use canny edge detection to perform blink detection. That is, canny edge detection allows the pupil to be found when the eye is open and a blink to be detected when it is not.

While their approach is published to determine accurate eye features in 98% of test frames, a frame rate of only three frames per second is achieved.

Many other techniques exist. The author directs the interested reader to [11, 5, 30, 28, 23].

3 Techniques and Algorithms

3.1 Marker Based Head Tracking

This section describes two techniques investigated for the purpose of ascertaining an actors head pose in 3-space by use of fiducial markers. The first is an approximation technique using three round reflective markers. The second is a technique using a single ARToolkit marker and using ARToolkit to derive pose information.

3.1.1 3 Points to a Plane 3D Pose Estimation

Physical System

We wish to develop a pose estimation system consisting of a single camera and 3 spherical, reflective fiducial markers. A overhead view of the system is given in figure 3.1.

Two markers are placed on the left and right side of the actors head at a distance of $a$mm appart, perpendicular to the ears. A third marker is placed on the same horizontal place as the first two markers in front of the actor at a perpendicular distance of $b$mm. The markers themselves are a fluorescent orange for high reflectance and are 38mm in diameter. A carbon fiber head rig (see figure 3.2(a)) was constructed to hold the markers. The full weight of the head rig is 231g.

A single camera is used for determining head pose. The camera used was a microDV digital video camera recording at 720 x 568 pixels. Images were transfered to PC for processing via firewire. The camera is placed on the same horizontal place as the three markers in such a way that the front marker is between the camera and the line joining the first two markers. The distance between the front marker and the camera is $d$mm.

Marker Segmentation

Given a frame of captured video, our first step toward identifying pose based on fiducial markers is to identify all markers in the scene. Specifically, we must develop functionality that, given an image, returns, for each marker, the centroid or that marker in the 2D image space/plane.

The technique used in this system is similar to that described by Grant et al. [9]. However, the algorithm described by Grant returned the largest connected component as only one marker of each colour would be in the scene at any given time. Our application attempts to generalise this idea so that many markers of many colours can be detected. Should m markers of a particular colour be sought, the center of the m largest rectangles (below some user defined upper bound) are returned by the algorithm as the centroids of the markers. If $n < m$ components are found by the algorithm, the remaining $m - n$ rectangles are returned as having area zero to show only n markers were found.

Given a source frame I and a HSV colour space range $(H_{min}, H_{max}, S_{min}, S_{max}, V_{min}, V_{max})$, we define each pixel of the mask image M as shown in equation 3.1;

$$M_{x,y} = \begin{cases} 1 & (H_{min} \leq I_{H(x,y)} \leq H_{max} \cap \\ & S_{min} \leq I_{S(x,y)} \leq S_{max} \cap \\ & V_{min} \leq I_{V(x,y)} \leq V_{max}) \\ 0 & else \end{cases} \tag{3.1}$$

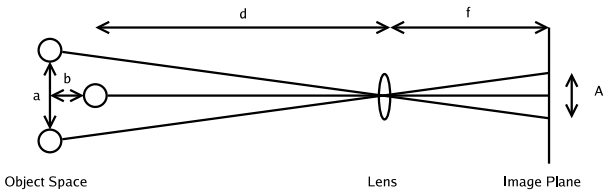


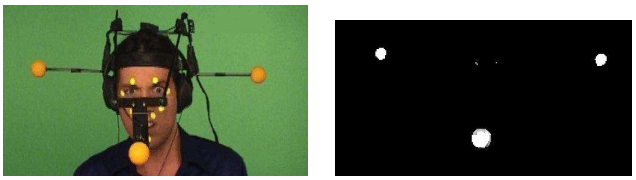
Figure 3.1: A symbolic top down view of the motion capture environment

At this stage the mask should isolate the markers in the frame but will also likely contain large amounts of spurious noise due to pixels falling within the marker colour range. Morphological operators are used to remove many of these small components. That is, erosion (see equation 3.2) is performed with a 3x3 kernel to remove small components, followed by a dilation (see equation 3.3) with a 3x3 kernel to rebuild the remaining components to approximately their original size.

$$E_{x,y} = \min(M_{x',y'} | x \leq x' \leq x+2, y \leq y' \leq y+2) \tag{3.2}$$

$$D_{x,y} = \max(M_{x',y'} | x \leq x' \leq x+2, y \leq y' \leq y+2) \tag{3.3}$$

A before and after example of this process is given in figure 3.2.



(a) Unprocessed video frame. (b) Mask image showing segmented markers.

Figure 3.2: An example of HSV based fiducial marker segmentation.

Point Correspondence

Having an accurate segmentation algorithm in place to provide centroids of each coloured marker, we set about assigning particular found centroids to specific markers. That is, segmentation gives us simply a list of markers. Point correspondence is the task of ascertaining which found centroid corresponds to, in our case, the front, left and right marker.

Point correspondence is a difficult problem. General solutions involve storing models of the marker positions in object space within the application. This way, the application can compare centroid positions on the image plane with the stored model to see which association is most likely.

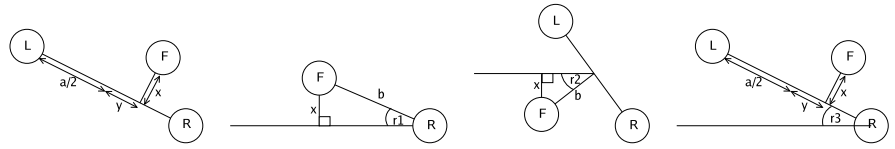
We have several advantages in performing motion tracking on a body on which only the head (and not shoulders or the rest of the body) is moving. Several assumptions can be made. Firstly, the head will never appear up-side-down with respect to the camera. Secondly, the head will always be, with a vertical and horizontal error of θ° for some θ, facing the camera.

We consider the option of simply assigning the left most centroid to the left marker, the right most centroid to the right marker and the remaining centroid to the front marker. Clearly such a method is not a general solution. However, given our assumptions we wish to discover whether such a simplistic approach is actually inadequate. An experiment in chapter 4 attampts to answer this question.

Pose Estimation

Pose estimation is the process of taking the 2d position of the left, right and front markers as given by the correspondence phase, and calculating, in our case the rotation of the head around the x, y and z axes. We present here an approximation algorithm for the solution to the head pose estimation problem and consider in section 4.1 the accuracy of this approach.

Consider the diagram in figure 3.1 along side the four diagrams in figure 3.3.



(a) 2D Image plane measurements (b) Side view in 3d showing how x implies x-axis rotation (c) Top view in 3d showing how y implies y-axis rotation (d) Front view in 3d showing how line L to R implies z-axis rotation

Figure 3.3: Head rotation pose estimation.

Angle of Rotation about X-axis We use the value of x in figure 3.3(a) to calculate α, the angle of rotation around the x-axis, as shown in figure 3.3(b) and equation 3.4.

$$\alpha = sin^{-1}\frac{x}{b} \tag{3.4}$$

Angle of Rotation about Y-axis We use the value of y in figure 3.3(a) to calculate β, the angle of rotation around the y-axis, as shown in figure 3.3(c) and equation 3.5.

$$\beta = sin^{-1}\frac{y}{b} \tag{3.5}$$

Angle of Rotation about Z-axis We use the line from the left and right marker in figure 3.3(a) to calculate γ, the angle of rotation around the z-axis, as shown in figure 3.3(d) and equation 3.6.

$$\gamma = tan^{-1}\frac{R_y - L_y}{R_x - L_x} \tag{3.6}$$

3.1.2 ARToolkit Marker Pose Estimation

During the main rendering loop of an ARToolkit application, we can obtain an OpenGL style 4×4 transformation matrix that represents an ARToolkit marker's position with respect to the camera via the ARToolkit function `arGetTransMat(...)`. Such a matrix has the form shown in equation 3.7;

$$\begin{pmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.7}$$

Such a transformation matrix contains a 3×3 rotation matrix R in the top left corner. Imagine R is the result of rotating the marker with respect to the camera α degrees around the x axis, β degrees around the y axis and γ degrees around the z axis. R can be reached from these so called Euler angles as shown in equation 3.11 below;

$$R = R_x(\alpha)R_z(\gamma)R_y(\beta) \tag{3.8}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{pmatrix} \begin{pmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & cos(\beta) & sin(\beta) \\ 0 & 1 & 0 \\ 0 & -sin(\beta) & cos(\beta) \end{pmatrix} \tag{3.9}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{pmatrix} \begin{pmatrix} cos(\gamma)cos(\beta) & -sin(\gamma) & cos(\gamma)sin(\beta) \\ sin(\gamma)cos(\beta) & cos(\gamma) & sin(\gamma)sin(\beta) \\ -sin(\beta) & 0 & cos(\beta) \end{pmatrix} \tag{3.10}$$

$$= \begin{pmatrix} cos(\gamma)cos(\beta) & -sin(\gamma) & cos(\gamma)sin(\beta) \\ cos(\alpha)sin(\gamma)cos(\beta) + sin(\alpha)sin(\beta) & cos(\alpha)cos(\gamma) & cos(\alpha)sin(\gamma)sin(\beta) - sin(\alpha)cos(\beta) \\ sin(\alpha)sin(\gamma)cos(\beta) - cos(\alpha)sin(\beta) & sin(\alpha)cos(\gamma) & sin(\alpha)sin(\gamma)sin(\beta) + cos(\alpha)cos(\beta) \end{pmatrix} \tag{3.11}$$

Using this result together with the matrix R supplied by ARToolkit, we can retrieve the angles α, β and γ that have been applied to the marker using the equations 3.12, 3.13 and 3.14 respectively, as follows;

$$\alpha = tan^{-1}(\frac{r_{21}}{r_{11}}) \tag{3.12}$$

$$\beta = tan^{-1}(\frac{r_{02}}{r_{00}}) \tag{3.13}$$

$$\gamma = tan^{-1}(\frac{-r_{01}}{\sqrt{a_{00}^2 + a_{02}^2}}) \tag{3.14}$$

3.2 Marker Based Face Feature Tracking

In a fiducial marker based face feature tracking system small reflective markers are placed in various positions on the face so that a system tracking these markers may gain movement information about the underlying face. Often high resolution infra-red camera track specially build IR reflective markers. As these cameras are often prohibitively expensive we consider a system using a cheap 640 x 480 pixel webcam. As with head pose marker tracking, segmentation of markers in the image plane is an imperative first step in the face motion capture pipeline. We consider three approaches to segmentation, test them experimentally (see section 4.2) and discuss them.

3.2.1 Software HSV Segmentation

Software HSV face marker segmentation works in a similar way to the head marker segmentation described in chapter 3.1. That is, a range is defined in HSV colour space within which markers should fall. Binary masks are created to show which pixels fall inside the HSV range and morphological operators are used to reduce noise. Finally, large regions possibly matching markers are identified using a connected component algorithm[9].

In our tests eight 1cm reflective coloured markers were placed on the face as shown in figure 3.4. The marker rig used in the head pose estimation system (see figure 3.2(a)) contains a small 640 x 480 colour webcam mounted facing the actors face directly behind the front head pose marker. Such a camera remains fixed with respect to the face and greatly reduces the complexity of face pose estimation. The segmentation of the face markers is completed in the same way as that for head markers described in chapter 3.1.

3.2.2 Hardware Camera Properties Segmentation

Modern webcams ship with drivers that allow a large amount of filtering on the camera itself before an image enters the main image processing system. We wish to gain an insight of whether such customisations can improve the segmentation portion of motion capture. Specifically, motion capture camera systems are usually working on IR light so that lighting causes less problems. As such can we simulate the greyscale result usually achieved by IR cameras by varying our webcams parameters.

Figure 3.4: Source image frame for regular HSV segmentation

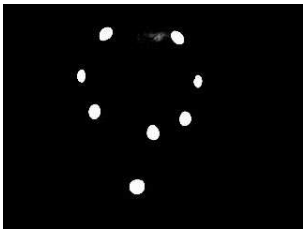


Figure 3.5: Source image frame for camera hardware filtering HSV segmentation

The camera used in the head rig for face capture was a Logitech QuickCam for Notebooks Pro [1] set to capture 640 x 480 pixels. All driver settings were set to default except the following. Saturation was set to zero and contrast was set to maximum so that the image became a binary black and white image. Further, brightness was set to approximately 5% so that only the highly reflective markers would appear on the image. While some noise would occure from reflective teeth, whites of the eye, and reflections off skin, we hypothesies that preprocessing the image in this way greatly increases the ability of the HSV segmentation described in [9] to correctly segment markers.

3.2.3 Ultra Violet HSV Segmentation

While we expect the hardware camera properties segmentation technique will result in cleaner segmentation, it has the disadvantage of losing all colour information. That is, it is not possible to detect the difference between a, for example, red and green marker. We consider the possiblity of, rather than using an infra-red system as many motion capture environments do, using a ultra-violet based system. For this experiment 11 fluorescent markers were applied to a subjects face. A standard 720 x 568 pixel microDV video camera was used to capture the markers. The face of the subject is illuminated by a single General Electric F20T12/BLB 24in UV bulb [2] producing UVa light at a wavelength of 368nm. The reflectance properties of difference materials is difference under UV light compared to regular light. Fluorescent materials modify UV light into visible light when reflecting it. Contrary to this, human skin does not so UV light reflected off skin remains invisible. By using such a light in our capture environment we find that the image plane contains only the markers which are emitting visible light, not the face and surrounding room that is reflecting UV light. Figure 3.6 shows a frame of video captured under this scenario.

3.3 Markerless Face Feature Tracking

Applications requiring the accurate identification of eye pupil position in a two dimensional image can be found in many areas, ranging from gaze based computer interfaces [4] to motion capture. Many possible solutions have been put forward to this problem including approaches involving

[1] http://www.logitech.com/index.cfm/products/details/NZ/EN,CRID=2204,CONTENTID=10561
[2] http://genet.gelighting.com/LightProducts/Dispatcher?REQUEST=COMMERCIALSPECPAGE&PRODUCTCODE=34747

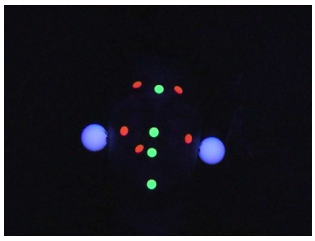


Figure 3.6: Source image frame for UV based HSV segmentation

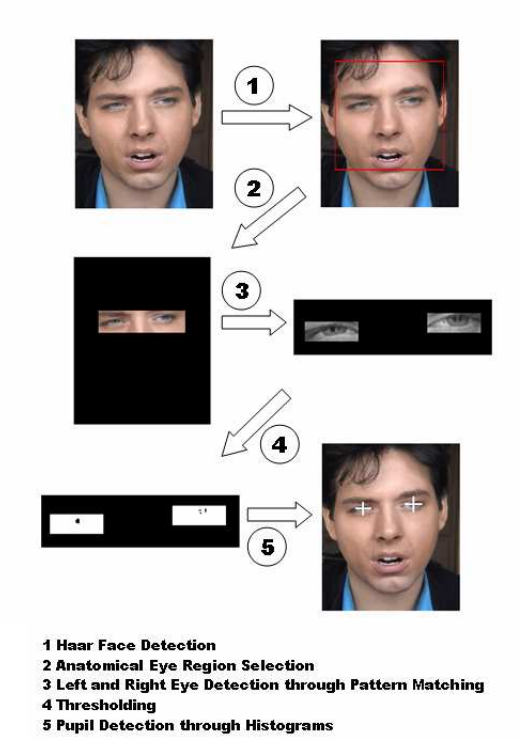


Figure 3.7: Hierarchical Methodology for Pupil Detection

Haar cascades, thresholding, Hough transforms, templates and pattern matching.

We present an algorithm, created out of several techniques, that recursively narrows the region of interest from the full frame to the pupils. Figure 3.7 outlines our technique and the separate steps are outlined in more detail in the sections below. We begin by using Haar-like features to detect the face region (section 3.3.1). Known anatomical proportions of the face allow us to further narrow the region of interest to the area around the eyes (section 3.3.2). Pattern matching using grayscale eye like images is now used to identify the left and right eye separately (section 3.3.3). Finally, we use thresholding and vertical and horizontal histograms to find the location of the pupil within the eye (section 3.3.4).

3.3.1 Haar Face Detection

The use of statistical cascade classifiers based on Haar-like features have a long history in object detection, particularly in face detection[31, 22, 16, 24, 11]. So widespread is their use that many computer vision APIs, such as OpenCV [20], ship with example Haar classifiers and the functions to access them. The OpenCV implementations of Haar classifiers were used by our system.

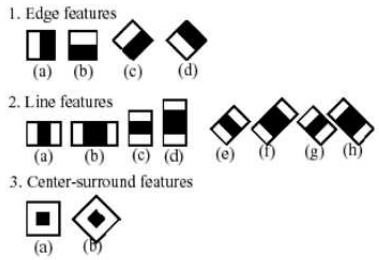


Figure 3.8: Extended Set of Haar-like Features used in OpenCV's Implementation of Face Detection (taken from [22])

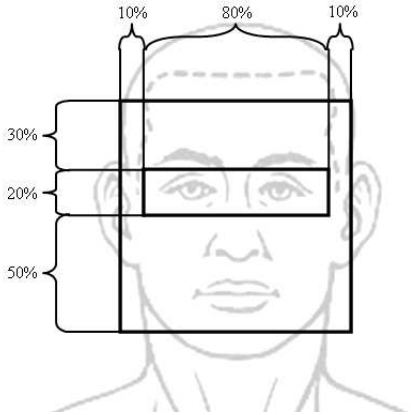


Figure 3.9: Anatomical percentages used to isolate the eye region.

Such techniques involve training a cascade of boosted tree classifiers via several thousand positive sub-images (often 24x24 pixel) and negative images. In the OpenCV implementation, simple features are described by a number of templates as shown in figure 3.8 and described in [22]. Detection involves moving the classifier in a sliding window around the image in an attempt to find regions of the image that match the classifier.

Haar cascade face detection makes up the first step of our algorithm. This step narrows the region of interest in the frame to the face of the subject (see figure 3.7 step 1). Subsequent steps below only consider this localised region.

3.3.2 Anatomical Eye Region Selection

An assumption in our system, that the above Haar cascade relies upon, is that the user is facing the camera with reasonably little rotation around the three primary axes. Such an assumption allows us to further narrow the region of interest using known anthropometric proportions of the eyes within the face region. Figure 3.9 shows the proportions used to isolate the eye region. Several different ratios were considered. It was found that the 1:8:1 horizontal and 3:2:5 vertical ratios defined the smallest eye region that robustly segmented the eye region given an image with a successful Haar based face segmentation.

3.3.3 Individual Eye Detection Through Pattern Matching

By dividing the eye region (see 3rd image of figure 3.7) in half with a vertical line we have relatively (to the original image) small regions of interest containing the left and right eye. If we make the assumption that these regions contain only the eye we may step directly onto the histogram technique described in section 3.3.4. However, in many cases these eye regions also contain part

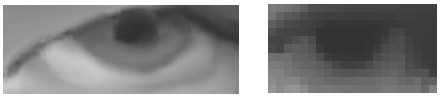
or all of the eyebrows. Since, commonly, eyebrows are dark in colour, if we do not remove them from the region of interest the histogram technique will fail.

We further narrow the region of interest for the left and right eye by using a generic, low resolution, grayscale eye image. An attempt is made to find the area in the eye region which best matches this pattern. The process takes place with a grayscale version of the original frame.

Figure 3.10 shows the images used as patterns throughout investigations of this technique. Pattern 3.10(b) was eventually used in all experiments.

We wish to find a rectangle of pixels, in the $s \times t$ eye region I, that best matches the $n \times m$ pattern P. If we define the top left corner of this best match to be C and put $X_{x,y}$ as the pixel of some image X intersected by the x^{th} column and y^{th} row, then the best match in the eye region is defined as $C = I_{x,y}$ where $0 \leq x \leq s - n$ and $0 \leq y \leq t - m$ and $E(x, y)$ is the minimum across all x, y as given by equation 3.15.

$$E(i,j) = \sum_{k=0}^{n} \sum_{l=0}^{m} |I_{i+k,j+l} - P_{k,l}| \tag{3.15}$$



(a) (b)

Figure 3.10: Examples of grayscale eye patterns.

3.3.4 Pupil Detection Through Row vs Column Histograms

Having narrowed the region of interest to two small rectangles representing the left and right eye (see image 4 of figure 3.7), we consider a technique for finding the pupil. As the left and right eye regions do not contain eyebrows, we may make the assumption that the pupil is the darkest region in the eye. As such, we threshold the image into a binary image so that the pupil area appears black or 'on', while most other areas are white or 'off'. We count the number of 'on' pixels in each row and column and consider the pupil to be at the intersection of the column and row with the most 'on' pixels. Formally, given an $n \times m$ eye region R, we define the pupil location P as;

$$P = \left(\underset{i}{\text{argmax}} \left(\sum_{j=0}^{m} R_{i,j} \right), \underset{j}{\text{argmax}} \left(\sum_{i=0}^{n} R_{i,j} \right) \right)$$

This concept is clarified in figure 3.11.

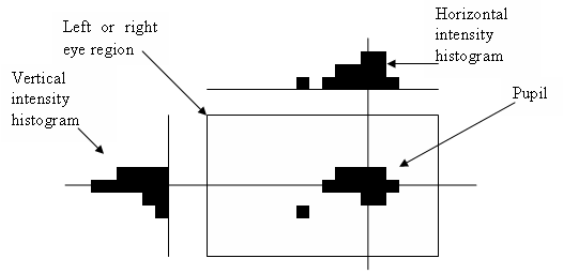


Figure 3.11: Vertical and horizontal histograms showing the location of the pupil.

4 Analysis

The primary goal of this research was to identify algorithms in the motion capture pipeline that could be optimised. In this chapter we perform six experiments that evaluate the techniques introduced in the previous chapter.

4.1 Marker Based Head Tracking

4.1.1 3 Points on a Plane 3D Pose Estimation

Marker Segmentation

We wish to perform an experiment to ascertain the suitability of the static HSV marker segmentation technique similar to that described in [9]. To do this we use a similar experimental technique as that used by Grant et al. Three video sequences are constructed, one with 10 frames and two with 1000 frames. All sequences are shot with a static 720 x 568 pixel camera capturing a person's head with a marker rig (see figure 3.2(a) attached. The person is moving throughout the three sequences, but always satisfying the assumptions given above.

The first step of the experiment is to set the parameters for $(H_{min}, H_{max}, S_{min}, S_{max}, V_{min}, V_{max})$ based on the first 10 frame video sequence so that the markers in that sequence are correctly segmented. The two 1000 frame sequences are now processed. For each frame the system shows graphically where it believes the three markers are on the 2d image plane.

Table 4.1 shows the percentage of correct hits attained by the system across the two 1000 frame videos. A hit is deemed correct if the 2d point returned by the system falls within the 2d image plane area that represents the marker.

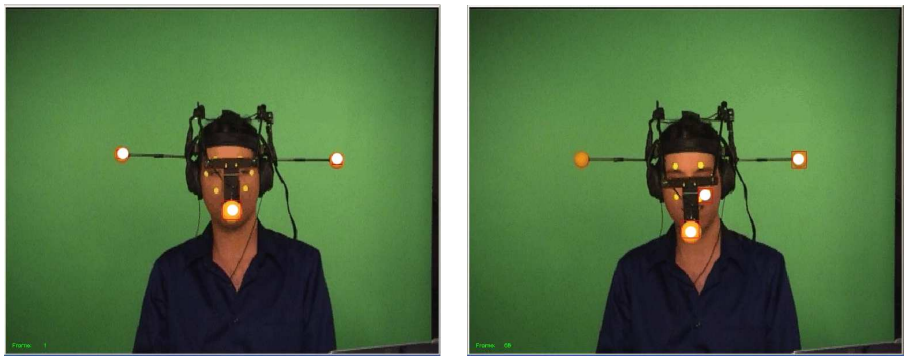
	Correct Hits (%)	Max Incorrect Sequence
Sequence 1	100%	0
Sequence 2	99.9%	3
Overall	99.95%	3

Table 4.1: Results for the Marker Segmentation Experiment

Examples of correct and incorrect segmentations are given in figure 4.1. Also given in table 4.1 are results for the longest sequence of incorrect hits. The negative effects of incorrect hits can be reduced by post processing with a Kalman filter or a particle filter. Such filters use probabilistic properties of the sequence to determine if the measured position is likely. However, long sequences of incorrect measurements will confuse particularly the Kalman filter. As such, keeping sequences of incorrect hits as short as possible is preferable.

Point Correspondence

Assuming a person being tracked stays within the assumptions of our system, we know that the simple point correspondence technique described in section 3.1.1 will work correctly. We constructed a system with $a = 600mm$, $b = 300mm$ and $d = 2000mm$. We wish to make an empirical measurement showing how likely it is for an actor, during a typical performance, to move outside this range. To measure this four 1000 frame video sequences were taken of an



(a) Segmentation Correct Hit (b) Segmentation Incorrect Hit

Figure 4.1: Examples of marker segmentations

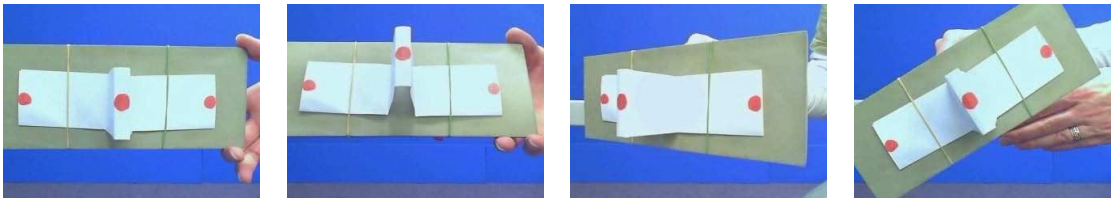
actor performing a song and four 100 frame video sequences were taken of an actor in general conversation. The number of frames where the actor violated the y-axis rotation constraint were counted and are given, as a percentage, in table 4.2.

	Percentage of frames within constraints
Sequence 1	100%
Sequence 2	100%
Sequence 3	100%
Sequence 4	100%
Sequence 5	100%
Sequence 6	80%
Sequence 7	87%
Sequence 8	100%
Overall	95.875%

Table 4.2: Results for the Point Correspondence Experiment

Pose Estimation

The three point pose estimation technique described in 3.1.1 was implemented and tested with three points set at various angles. Figure 4.2 shows four such rotations. We note that all angle estimations are within 2.6° of the actual angle save for one angle around the y-axis (see figure 4.2(c)).



(a) x=0(-0.25), y=0(1.51), z=0(0.99) (b) x=30(29.13), y=0(2.18), z=0(1.23) (c) x=0(-1.64), y=30(41.77), z=0(1.84) (d) x=0(-2.51), y=0(-0.68), z=-30(-31.09)

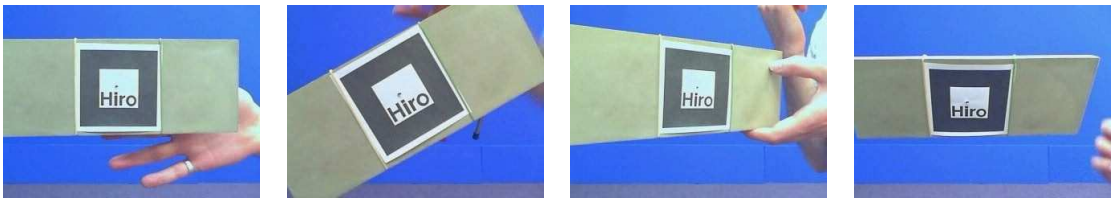
Figure 4.2: 3 Point Based Pose Estimation Examples (Key:Actual(Measured))

Average pixel error	3.19
Maximum pixel error	12.5
Minimum pixel error	1
Average Frames Per Second	4.75

Table 4.3: Results for experiment on 285 frame test video.

4.1.2 ARToolkit 3D Pose Estimation

We wish to perform an experiment to ascertain the suitability of the algorithms within ARToolkit for 3D pose estimation for motion capture. As such we implement and test the ARToolkit pose estimation technique with an ARToolkit marker set at various angles. Figure 4.3 shows four such rotations. We note that all angle estimations are within 7.3° of the actual angle.



(a) x=0(1.71), y=0(2.29), z=0(-0.57) (b) x=0(5.73), y=0(4.58), z=-30(-30.37) (c) x=0(5.73), y=30(29.79), z=0(-4.01) (d) x=-30(-37.24), y=0(4.58), z=0(-2.86)

Figure 4.3: AR Based Pose Estimation Examples (Key:Actual(Measured))

4.2 Marker Based Face Feature Tracking

We wish to perform an experiment that compares the three segmentation techniques discussed in 3.2 to see whether any are significantly more robust than others. Six video sequences are constructed, one with 10 frames and one with 1000 frames for each technique. The first step of the experiment is to set the parameters for $(H_{min}, H_{max}, S_{min}, S_{max}, V_{min}, V_{max})$ for each technique based on the first 10 frame video sequence so that the markers in that sequence are correctly segmented. Each 1000 frame sequence is now processed, once for each technique. For each frame the system shows graphically where it believes the facial markers are on the 2d image plane. Figure 4.4 shows the percentage of correct hits attained by the system across the 1000 frame video for each technique. A hit is deemed correct if the 2d point returned by the system falls within the 2d image plane area that represents a marker.

4.3 Markerless Face Feature Tracking

We tested our implementation of the hybrid pupil tracking approach approach over 285 frames of a sample 320 x 240 avi video on a 2.8GHz Intel Pentium 4 with 448MB of RAM. The video showed a subject moving their eyes left, right, up and down as well as moving their head left, right, up and down with respect to the camera. No artificial lighting was used in the recording of the video though the room was well lit with natural sunlight diffused by cloud. The pupil locations in all 285 frames were first manually recorded and these values were compared with the values returned by our algorithm. Examples of correct and incorrect pupil segmentations are shown in figure 4.5. Results for the test video are shown in table 4.3.

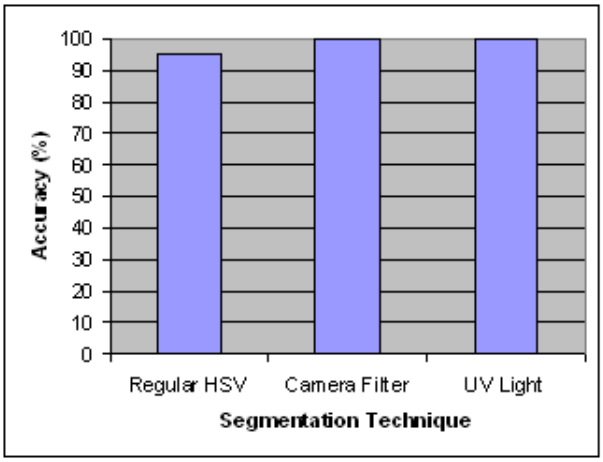
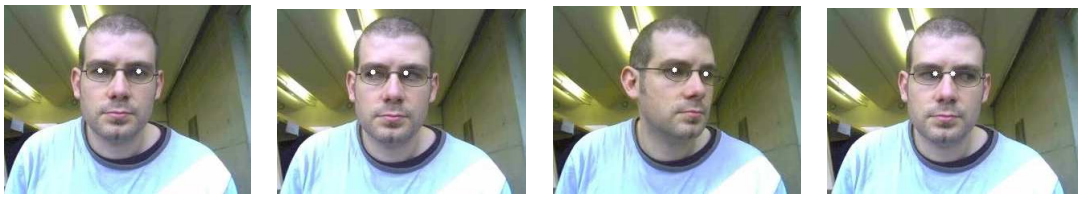


Figure 4.4: Comparison of accuracy of three face marker segmentation techniques.



(a) Correct (b) Correct (c) Correct (d) Incorrect

Figure 4.5: Examples of three successful and one unsuccessful pupil segmentation.

5 Discussion

5.1 Head Pose Estimation Segmentation Technique

Chapter 4 begins with an experiment testing the robustness of a HSV space, fiducial marker segmentation technique presented by Grant et al. [9]. The technique worked very robustly, making no errors for the first 1000 frame test video and achieving an accuracy of 99.9% for the second 1000 frame video. Grant et al. present an average 98.7% accurancy for their system. Our results, at an average of 99.95%, are slightly higher. The difference is most likely due to differences in the types of markers and differences in how the experiment was run. Grant's markers were coloured objects held by the actor and were sometimes partially obscured by the hand holding them or by another part of the actor. No such problems occur in our system as the markers are placed so that non are obscured if the actor satisfies the assumptions of the system.

Briefly mentioned in chapter 4 is the maximum length of incorrect results observed. Due to marker obscuring etc, it is unrealistic that all markers will appear in frame all the time. To combat this problem algorithms like Kalman filtering and Particle filtering use the statistical properties of the motion sequence to estimate the correct location of a marker when it is either obscured or if the system's confidence in its location is too low. Such filters may get confused and output significantly incorrect approximations if long sequences of incorrect segmentations occur. We report a maximum incorrect sequence of just three frames in chapter 4, which is well within the bounds of operation for both Kalman and Particle filters.

All experiments for HSV segmentation were carried out in a studio environment with soft diffuse lighting and uncluttered backgrounds. This is the HSV segmentations greatest dissadvantage. Non ideal conditions with constantly varying and harsh light sources will vastly change the cameras perceived colour of the marker. As such, the HSV colour ranges need to be suitably large to accommodate these changes. This inturn increases the chances of false positives being reported. Grant et al's dynamic segmentation technique was designed to combat this problem but did not provide an addequate solution.

5.2 Point Correspondence

In chapter 3 we present a relatively simplistic solution to the point correspondence problem. The solution relies on the actor not rotating their head around the y-axis more than a preset number of degrees. The experiment outlined in chapter 4 attempts to ascertain whether such a requirement is realistic. Interestingly, results show that the probability of the actor violating the y-axis contraint changes depending on the type of activity the actor is engaged in. The chance of the actor breaking the constraint when singing is considerably less that while engaged in general conversation.

To find an explanation for why this occurs we return to the original video footage. It was hypothesised that singing would cause more contraint violations. Looking a the video, though, we noted that the actor tended to sing to the camera, as if they were performing on television. Contrary to this, when engaged in conversation, the actor tended to move his head around more as if speaking with a group of people. Clearly the type of activity being captured needs to be well understood before decided to solve point correspondence in our way.

5.3 3 Point vs ARToolkit Pose Estimation

Two pose estimation techniques has been described for finding the rotation of a rigid object in 3 space. Our hypothesis was that the ARToolkit pose estimation would be generally more accurate than the 3 point approximation technique but suffer from occasional jitter that would create large errors. Contrary to this hypothesis the 3 point approximation technique provided more accurate and more stable angle estimations. An interesting question that we have as yet been unable to answer is why did the ARToolkit solution perform relatively poorly. The problem is not due to the core pose estimation algorithms within ARToolkit as while ARToolkit was being asked for angles it was also asked to place a 3D object on the marker, were ARToolkits pose estimation to be as erroneous as our results suggest, the 3d object would be placed incorrectly, which it is not. So we have a problem either in how we are reading values from within ARToolkit, or how we mathematically manipulate those figures to produce angle figures. As the mathematics is relatively straightforward, we suspect the error lies in how we read figures from ARToolkit.

A second intersting point is why the 3 point pose estimation technique produces an error around the y-axis only. Once may first assume an error has been made in the calculation of this component, however, like with the ARToolkit calculations, the mathematics is relatively straightforward. More likely is that the technique itself does not approximation rotations around the y-axis well. Given the general success of the 3 point technique, a further investigation to improve its y-axis accuracy would be beneficial.

A further disadvantage of ARToolkit is that, being designed for augmented reality and as such, real time processing, it has a tendency to skip frames if it falls behind. In motion capture timing information for the resulting movements is given by the number of frames processed and the frames per second of the recording. If frames are skipped by ARToolkit timing information is corrupted. It has been commented that an option in ARToolkit exists to turn off frame skipping but despite a certain amount of searching, such an option could not be found.

5.4 Facial Marker Segmentation Techniques

For the purpose of marker segmentation for markers placed on the face we considered the basic HSV segmentation as well as two techniques for improving this technique, camera hardware filtering and UV lighting. While the basic HSV segmentation performed similarly to the results given by Grant et al., both improvements resulted in 100% accuracy.

Camera hardware filtering achieves its gain by converting the normal three colour image into what amounts to an intensity map. The highly reflective markers register on this intensity map where as the less reflective areas of the skin and clothing and background do not register, hence making segmentation trivial. The dissadvantage here is that by modifying the colour image to an intensity map we lose all colour information. That is, all coloured markers are registered identically and and indistinguishable on the intensity map. While many applications call for only one colour, multiple colours are sometimes useful for point correspondence issues.

Many systems use IR light to avoid the segmentation problems associated with natural light conditions. We consider a setup where only UV light is allowed to shine on the markers. UV light is naturally invisible as it is below the visible spectrum of light. However, some materials, when they reflect UV light, modify UV's wavelength so that it enters the visible spectrum. Markers coated in fluorescent colours have this property and were used in our experiments. Similar to the camera hardware filtering technique, the result is only the markers are visible on the image plane as the skin and background to not become visible in UV light. A further advantage, however, is that markers do not lose their colour information. The UV light is transformed into that part of the spectrum where the colour that reflected it resides. This is a distinct advantage over camera hardware filtering. The only clear disadvantage to UV light is its health concerns. UV light is catergorised into three brackets, UVa, UVb and UVc, from least to most dangerous. UVa is a type of light produced in large quantities by the sun.

5.5 Eye Pupil Tracking Hybrid Technique

We see from table 4.3 that our hybrid pupil tracking algorithm achieves an excellent average error of only 3.199 pixels. This is particularly promising when one considers the average eye width in the test video of 21 pixels. While not accurate enough for applications such as gaze based input, our technique would be well suited to post processed motion capture.

The maximum error measured between actual pupil and detected pupil was 12.5 pixels. Such a low error is due to the hierarchical nature of the algorithm. That is, should the final stage of the algorithm be unsuccessful at locating the pupil, a successful eye region detection by the previous step will have reduced the possible error. Of course, if the Haar face detection is the layer of the hierarchy that fails, the results will most likely be dramatic.

While 4.75 frames per second is higher than some approaches, it is still too low to be considered for real time applications of either motion capture or gaze based input. The low frame rate is due primarily to the Haar face detection which accounts for over 80% of processing time. Techniques exist that improve the performance of haar face detection to the point where it may be considered for real time applications.

6 Conclusion

We have investigated, implemented and evaluated six algorithms that attempt to solve three of the main challenges facing motion capture systems for head/face models. Two techniques to solve the 3D pose estimation problem were investigated. One technique, based upon 3 fiducial markers falling on a plane, and another, built upon the ARToolkit, were implemented and evaluated. The marker segmentation problem for face placed fiducial markers was also considered. We compared three techniques that address this problem, namely HSV colour space segmentation, camera hardware filtering and UV illumination segmentation. Finally, we proposed a hierarchical pupil tracking algorithm combining Haar face detection, anthropometric localisation, pattern matching and row vs column intensity histograms.

All systems were implemented and experiments were performed. Both pose estimation techniques showed promise in a variety of poses, though, unexplained errors also occurred. The three segmentation approaches were tested on four 1000 frame and four 100 frame videos containing inherently different movements. The simple HSV segmentation technique is accurate 95% of the time whereas both improvements give 100% accuracy. We identify inherent advantages and disadvantage for the three techniques. We tested the performance of our pupil detection algorithm on a 285 frame video displaying a variety of gaze directions. Our hybrid approach performs well, resulting in a very low pixel error and a reasonable frame rate.

6.1 Future Work

Both pose estimation algorithms would benefit from further research attempting to reduce the errors they currectly display. Investigations into the health side affects of the UV light system will answer whether such a system is realistically usable on humans. Further development of the pupil tracking system will tune it, addressing its limitations. Most notably, these include a frame rate that makes real time tracking difficult and high error results when the user blinks. Improving the speed of the Haar face detection algorithm (the slowest part of the system) and including a technique to detect a blink state will address these issues.

7 Publications

M. Schoo and R. Green, "A Hybrid Approach For Tracking Eye Pupils", in press, Image and Vision Computing New Zealand, 2006. (Full paper published in proceedings of international conference which was independently reviewed.)

Bibliography

[1] *Out of the Inkwell: Max Fleisher and the Animation Revolution*. University Press of Kentucky, 2005.

[2] M. Billinghurst and H. Kato. Collaborative mixed reality. In *In Proceedings of International Symposium on Mixed Reality (ISMR '99). Mixed Reality–Merging Real and Virtual Worlds*, 1999.

[3] M. Billinghurst, I. Poupyrev, H. Kato, and R. May. Mixing realities in shared space: an augmented reality interface for collaborative computing. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, pages 1641–1644, 2000.

[4] G. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, (2), 1998.

[5] M. Covell. Eigen-points: Control-point location using principle component analyses. In *Proceedings of the 2nd international Conference on Automatic Face and Gesture Recognition (FG '96)*, page 122, Washington, DC, USA, 1996. IEEE Computer Society.

[6] F.Raab. Magnetic position and orientation tracking system. *IEEE Trans. Aerospace and Electronic Systems*, pages 709–717, 1979.

[7] M. Gleicher. Animation from observation: Motion capture and motion editing. In *Computer Graphics, 1999. Invited paper to appear in a special issue year = 1999, pdf = /home/cosc/student/msc61/stage4/cosc460/papers/Gleicher.M Animation From Observation Motion Capture and Motion Editing.pdf, url = citeseer.ist.psu.edu/gleicher99animation.html,*.

[8] Taro Goto, Marc Escher, Christian Zanardi, and Nadia Magnenat-Thalmann. MPEG-4 based animation with face feature tracking. In *Computer Animation and Simulation '99*, pages 89–98.

[9] R. N. Grant and R. Green. Tracking colour movement through colour space for real time human motion capture to drive an avatar. In *Image and Vision Computing New Zealand*, 2004.

[10] Qiang Ji and Xiaojie Yang. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-Time Imaging*, 8(5):357–377, 2002.

[11] K. S. Jin, S. G. Cho, J. S. Lee, and J. J. Hwang. Real-time pupil detection based on three-step hierarchy. In *Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference on*, volume 2, pages 1318 – 1321, August 2004.

[12] A. Kapoor and R. W. Picard. Real-time, fully automatic upper facial feature tracking. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 8–13, May 2002.

[13] Ashish Kapoor and Rosalind W. Picard. Real-time, fully automatic upper facial feature tracking. In *Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, page 10, 2002.

[14] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *IWAR '99: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, page 85, Washington, DC, USA, 1999. IEEE Computer Society.

[15] A. G. Kirk, J. F. O'Brien, and D. A. Forsyth. Skeletal parameter estimation from optical motion capture data. In *Computer Vision and Pattern Recognition, 2005*, page 1185. IEEE Computer Society Conference on, June 2005.

[16] A. Kuranov, R. Lienhart, and V. Pisarevsky. An empirical analysis of boosting algorithms for rapid objects with an extended set of haar-like features. Technical Report MRL-TR-July02-01, Intel Technical Report, 2002.

[17] P. Malbezin, W. Piekarski, and B. H. Thomas. Measuring artootkit accuracy in long distance tracking experiments. In *Augmented Reality Toolkit, The First IEEE International Workshop*, 2002.

[18] N. Malik, T. Dracos, and D. Papantoniou. Particle tracking in three dimensional turbulent flows part ii: Particle tracking. *Experiments in Fluids*, 15:279–294, 1993.

[19] H. Nanda and K. Fujimura. Illumination invariant head pose estimation using single camera. In *Intelligent Vehicles Symposium*, pages 434– 437. IEEE, June 2003.

[20] OpenCV : Open Source Computer Vision Library. Home page. http://www.intel.com/technology/computing-/opencv/index.htm, visited on 19/9/2006.

[21] S. M. Platt and N. I. Badler. Animating facial expressions. In *SIGGRAPH '81: Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, pages 245–252, New York, NY, USA, 1981. ACM Press.

[22] J. Maydt R. Lienhart. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages 900–903, September 2002.

[23] K. Toyama R. S. Feris, J. Gemmell. Facial feature detection using a hierarchical wavelet face database. Technical Report MSR-TR-2002-05, Microsoft Research Technical Report, 2005.

[24] S. Romdhani, P. Torr, B. Schkopf, and A. Blake. Efficient face detection by a cascaded support-vector machine expansion. In *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, volume 460, pages 3283 – 3297, November 2004.

[25] Antonio Carlos Sementille, Luís Escaramuzi Lourenço, José Remo Ferreira Brega, and Ildeberto Rodello. A motion capture system using passive markers. In *VRCAI '04: Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, pages 440–447, New York, NY, USA, 2004. ACM Press.

[26] D. J. Sturman. A brief history of motion capture for computer character animation. In *SIGGRAPH94, Course9*, 1994.

[27] D. J. Sturman. Computer puppetry. *Computer Graphics and Applications*, 18(1):38–45, 1998.

[28] Y. Tian, T. Kanade, and J. F. Cohn. Dual-state parametric eye tracking. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, page 110, Washington, DC, USA, 2000. IEEE Computer Society.

[29] T.Moeslund and E.Granum. A survey of computer vision based human motion capture. In *Computer Vision and Image Understanding*, page 231268, 2001.

[30] V. Uzunova. An eyelids and eye corners detection and tracking method for rapid iris tracking. Master's thesis, Otto-von-Guericke University of Magdeburg, 2005.

[31] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages 511–518, 2001.

[32] L. Williams. Performance-driven facial animation. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 235–242, New York, NY, USA, 1990. ACM Press.